



Geneapp Manual

Version 1.0 - 2023

Contents

1. Introduction to DAS analysis	2
2. GeneAPP Overview.....	3
3. Use cases of Geneapp.....	10
3.1. UC1: Consulting the gene structure in GeneappExplorer	10
3.2. UC2: Performing a DAS analysis with GeneappScript.....	11
3.3. UC3: Exploring data from another software	12
3.4. UC4: Using Geneapp via Docker	13
4. Outputs of Geneapp	14
4.1. Figures generated by other software	15
4.2. GeneappExplorer Descriptive graphs	15
4.3. Analysis chart	17
4.4. GeneappExplorer Volcanoplot.....	18
4.5. GeneappExplorer Expression heatmap	19
4.6. GeneappExplorer Functional annotation chart.....	19
4.7. GeneappExplorer AS event structure graph	20
4.8. Tables.....	21
5. Developers information	22
6. About Geneapp	23
7. License	23
8. References	24
9. Abbreviations.....	24

1. Introduction to DAS analysis

Alternative splicing (AS) is the alternative processing of messenger RNA (mRNA) transcribed from nuclear genes, a transformative process facilitated by the spliceosome. It involves altering pre-mRNA sites, resulting in the generation of different isoforms from the primary transcript. This mechanism empowers eukaryotes, particularly in multi-exon genes, to enhance the diversity of their transcriptome and proteome, even with a limited number of genes. In this context, differential alternative splicing (DAS) refers to the observed variations in the alternative splicing patterns of mRNA transcripts between different conditions or states, typically comparing distinct biological samples. To identify DAS, biological replicates of the transcriptome are initially obtained and subsequently sequenced. The sequencing data is then either i) mapped to the genome (mapping) or ii) aligned to the transcriptome to quantify the isoforms. This analytical process heavily relies on bioinformatics, particularly during the final stages of data generation and result exploration.

However, the analysis of alternative splicing (AS) using high-throughput short reads comes with certain limitations:

- 1) The probability of detecting the AS example in a sample is influenced by factors such as the sample size. For instance, if a 4Gbp sample with 1M reads contains 2 AS of gene g, a 2Gbp sample has a 50% chance of capturing those 2 AS. Larger sample sizes generally yield more reliable results.
- 2) Even with a large sample size, challenges may arise. For example, the AS may not be present if it is not captured during sequencing if the event does not occur in the analyzed condition, if it lacks annotation, or if AS is not analyzed 'de novo' (using tools like rMATS).
- 3) When comparing samples from control (CTRL) and experimental (CASE) conditions with DAS, the control sample may contain both canonical and alternative splicing, while the case sample may have none.
- 4) The presence of paralogous genes in the genome can complicate the identification of the locus of the AS gene.

Given the substantial sequencing depth required for DAS analyses and acknowledging the historically high costs associated with generating extensive data volumes using high-throughput technologies, DAS analyses were less prevalent than the more common practice of conducting Differential Expression (DE) analyses. Consequently, it is plausible to encounter datasets where only DE analyses have been conducted. We can leverage such datasets to explore and uncover potential differential alternative splicing events by extending the scope of our analyses from differential expression. Plant-specific data repositories, including NCBI, Phytosome, Ensemble, and organism-specific databases

such as Solnetwork and Guavadb, offer diverse sources for selecting the requisite data for subsequent DAS analyses, whether sourced from NCBI directly or stored in a local folder.

GeneAPP is a user-friendly tool designed to visualize the analysis of deidentified AS events. Comprising three modules—GeneAPP Explorer, Script, and Server—it provides an accessible platform for exploring AS in genes annotated within the NCBI database using their gene ID. Currently, GeneAPP exclusively utilizes genes from the NCBI database.

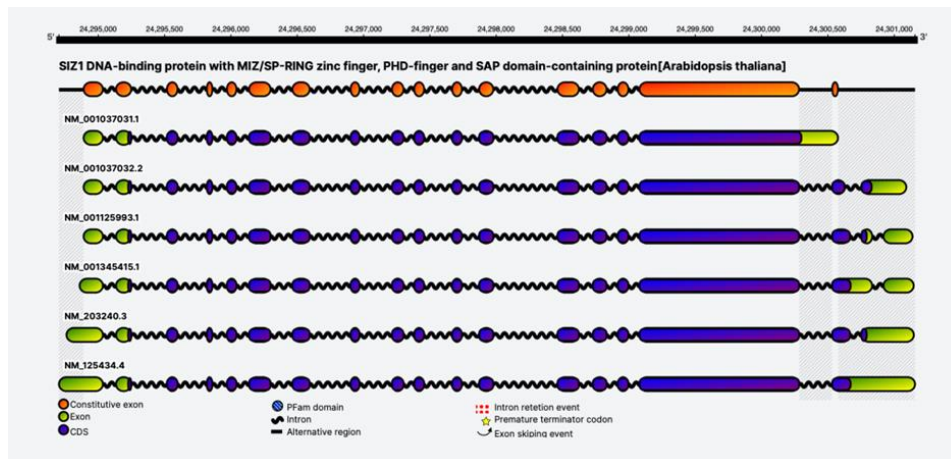
The primary function of GeneAPP is to identify AS patterns from RNAseq samples. The GeneAPP Script utilizes outputs from standard programs such as rMATS and 3DRNAseq, enabling the identification of gene Gene Ontology (GO) annotations, Pfam functional domains, and gene-link networks. This tool seamlessly integrates event-focused AS data from rMATS and transcript-focused information from 3DRNAseq. This integration facilitates the analysis of differential AS events, along with the prediction of new isoforms. The primary objective of GeneAPP is to aid users in selecting targets for experimental validation, given the identification of numerous genes with AS through various approaches.

divided into two sections: 1) Instructions for GeneAPP Usage - In this section, the steps for optimal utilization of GeneAPP will be detailed, ranging from data generation to result download. 2) Developer Section for Enhancing GeneAPP - This section provides instructions for developers interested in contributing to the improvement of the tool.

2. GeneAPP Overview

How to visualize the gene structure:

In this illustrative example, we will utilize a portion of the raw data sourced from an experiment available at the National Center for Biotechnology Information (NCBI) about *Arabidopsis thaliana*. To visualize the structure of a gene with annotation available on NCBI, the user can access the web application by providing the GeneID as indicated in the following URL: <https://gene.mikeias.net/gene?id=836163>, where in this case, the GeneID is 836163. After loading the data through the NCBI API, a graphical representation of the gene structure will be presented to the user, as shown in the figure below:



At the top, a scale is displayed indicating the gene's position in the genome sequence, with the 5' or anti-sense presented to the left of the bar, and the 3' to the right. Below the bar, the gene's name is shown based on its annotation obtained from NCBI. Beneath the name, the gene is represented as a consensus structure of its annotated isoforms. Wavy regions represent constitutive introns, orange sections denote constitutive exons, and dashes depict alternative regions in the gene. Below the consensus gene structure, annotated isoforms for the gene will be presented. In this example, there are six annotated isoforms for this gene. Each isoform is depicted with its exonic regions as a rounded green rectangle, shaded in purple in coding DNA sequences (CDS). If there is a recognized Pfam protein domain in the coding region, it will be displayed as a dashed light blue overlay where the overlap occurs.

How to generate AS data with Geneapp?

There are two ways to generate AS data with Geneapp: by importing previously executed data using GeneAPP Server or by generating the data after running GeneAPP Script. After defining the dataset, the next step involves utilizing software to unveil DAS and subsequently consolidating the results for importation into GeneAPP. GeneAPP is equipped to explore outcomes from both event-based software, such as rMATS, and isoform-based software like 3DRNASeq. To facilitate the execution of these analyses, which often necessitate numerous command lines for environment configuration and program execution, we have crafted the 'as_data_gen.sh' pipeline within a Bash script. This versatile script can be obtained and run on diverse platforms, including a local machine with a LINUX OS, cloud-based environments, clusters, or, as exemplified in this instance, on Colab. Users can execute the script by passing the arguments detailed below. A screenshot in the subsequent image illustrates the code to be entered in Colab. For testing purposes, users can clone this notebook. It's crucial to highlight that while the script was devised to assist users, it should be adapted based on the unique requirements of each experiment. Additionally, each software employed by the scripts carries its licensing requirements, and users are obligated to adhere to them under their responsibility.

```

[ ] ! curl -s https://raw.githubusercontent.com/MiqueiasFernandes/GeneAPP/main/as_data_gen.sh | bash -s - \
'https://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/001/735/GCF_000001735.4_TAIR10.1/GCF_000001735.4_TAIR10.1_genomic.f
'https://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/001/735/GCF_000001735.4_TAIR10.1/GCF_000001735.4_TAIR10.1_genomic.g
'https://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/001/735/GCF_000001735.4_TAIR10.1/GCF_000001735.4_TAIR10.1_cds_from
/content/drive/MyDrive/Doutorado/geneapp/completo/data/' \
SRA_ARG1='--minReadLen' SRA_ARG2='150' SRA_ARG3='-x' SRA_ARG4='999999' \
SRR21411875,CR1,WILD SRR21411876,CR2,WILD SRR21411877,CR3,WILD \
SRR21411881,MT1,MUTANT SRR21411882,MT2,MUTANT SRR21411883,MT3,MUTANT \
RMATS_ARG1='--readLength' RMATS_ARG2='150' KEEP_TMP \
1>/content/drive/MyDrive/Doutorado/geneapp/completo/log.t
2>/content/drive/MyDrive/Doutorado/geneapp/completo/err.t

```

- 1) In this line, the script is obtained directly from the repository to run.
- 2) The first argument is the location of the genome; you can pass the URL for it to be downloaded or the path to the FASTA file containing the nucleotide sequences of the genome on the computer where the script will be executed
- 3) Now, specify the path to the GTF annotation file. If a GFF file is present in the same directory, it will be automatically downloaded. Their names must match, differing only in the extension (.gtf / .gff).
- 4) Finally, the organism data requires the inclusion of transcript sequences. We recommend utilizing the CDS sequences for this purpose.
- 5) Specify the directory path to persist the results; in Colab, you can connect to Google Drive and inform the path within it.
- 6) Specify the arguments for the sra-tools. In this example, I am restricting the reads up to spot 999,999. For this test analysis, downloading the complete FASTQ file is unnecessary.
- 7) Provide each of the samples in the pattern RUN, NAME, FACTOR with a space.
- 8) Define the parameters to pass to the rMATS programs. In this instance, we are specifying the mandatory parameter --read-length with the value 150. Include other optional parameters in the script; for instance, the argument KEEP_TEMP indicates that the temporary folder should not be deleted.
- 9) These last two arguments are optional.

The steps involved in preparing the environment, downloading raw data, quality control, mapping, and annotation are resource-intensive and may require several days to complete; in this example, the process takes approximately 5 hours. Considering the substantial time investment and the fact that Colab recycles the environment, leading to the deletion of all data in the absence of user interaction, the script is designed to resume from the point of interruption for each step/sample. Processed data is retained in the output directory. To facilitate the reproduction of this use case, we recommend connecting the notebook to Google Drive and directing the output folder to a location within Google Drive. Alternatively, users should be vigilant in monitoring the script's progress if executed without connecting to Google Drive. If users attempt to reproduce the script on a local machine, they should be mindful of resource limitations, including RAM, DISK, and system settings. The script attempts to perform certain tasks in administrator mode, necessitating execution in a virtual machine or subsystem to avoid interference with current settings. It is not recommended to run it on localhost due to resource demands (>8GB RAM/>10GB disk).

Multiple outputs, execution logs, and data for importation into Geneapp will be generated by various programs. Among the files saved in the output folder, the most crucial are to3d.zip, rmats_out.zip, and results.zip. For a comprehensive quality control overview for each sample, refer to the file in the results folder, generated by multiqc.

The mapping rate in the genome can be found in the summary.txt file saved in the output folder.

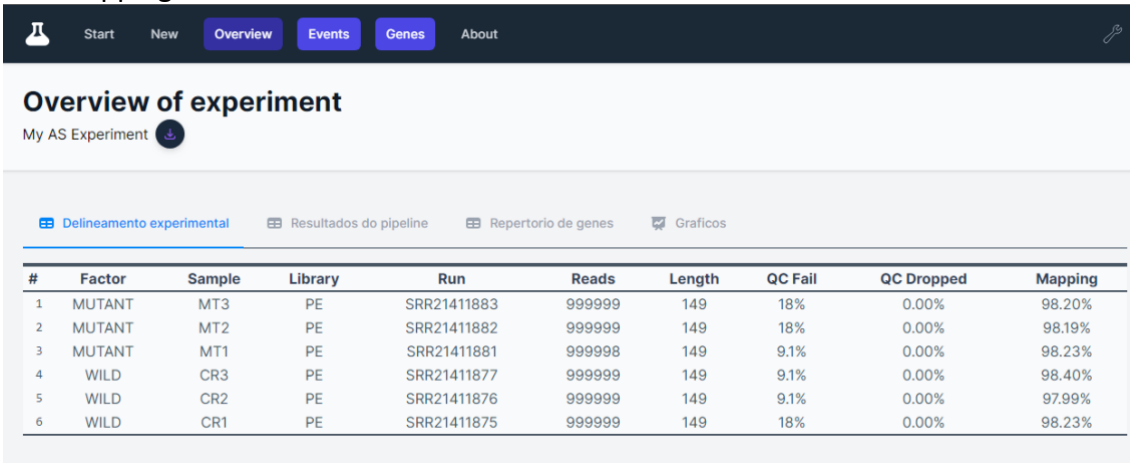
In the compressed file rmat_out.zip, all outputs from rMATS and MASER, including some graphics, are stored. Similarly, in the compressed file to3d.zip, all files necessary to execute the analyses on the web server of 3DRNaseq are stored. Since the script also runs 3DRNaseq, the results of a standard analysis in 3DRNaseq are included in this file. It is important to note that program parameters must be customized for each analysis. In this use case, default parameters for the programs were mostly utilized. Upon extracting the file import_geneapp.zip, users gain access to the Geneapp folder containing the requisite files for analysis within the Geneapp application.

How to analyze the generated data?

After the successful execution of the as_data_gen.sh pipeline (GeneAPP Script), users can access the import_geneapp.zip file. This file should be extracted before being loaded into the NEW page for exploration within GeneAPP. An example of this compressed file, generated using sample data in Colab, is available for download through the 'Download data sample' button. Upon successful data import into GeneAPP, and assuming all data has been processed, the user will have access to the following features:

Overview: page with data from the execution of AS analyses

Experimental design: Table with the list of processed samples, quality control metrics, and mapping.



#	Factor	Sample	Library	Run	Reads	Length	QC Fail	QC Dropped	Mapping
1	MUTANT	MT3	PE	SRR21411883	999999	149	18%	0.00%	98.20%
2	MUTANT	MT2	PE	SRR21411882	999999	149	18%	0.00%	98.19%
3	MUTANT	MT1	PE	SRR21411881	999998	149	9.1%	0.00%	98.23%
4	WILD	CR3	PE	SRR21411877	999999	149	9.1%	0.00%	98.40%
5	WILD	CR2	PE	SRR21411876	999999	149	9.1%	0.00%	97.99%
6	WILD	CR1	PE	SRR21411875	999999	149	18%	0.00%	98.23%

Pipeline results: Table with metadata of files used in the analyses and main numbers of analysis results.

Delineamento experimental Resultados do pipeline Repertorio de genes Graficos					
Step	Tool	Factor	Sample	Property	Value
Pre processamento				Genome Size	119Mpb
Pre processamento				Cromossomes	7
Pre processamento				Total genes	38319
Pre processamento				Genes coding	27561
Pre processamento				AS Genes coding	10695
Pre processamento				Transcripts	48265
Pre processamento				Transcriptome length	62Mpb
Pre processamento				Transcripts coding	31398
Pre processamento				AS Genes transcripts coding length	44Mpb
Pre processamento				RNASeq reads	12.0mi
Pre processamento				RNASeq read length	149
Pre processamento				Transcritome depth	28.8x
Align	Hisat	MUTANT	MT3	Mapping Genome	98.20%
Align	Hisat	MUTANT	MT3	Mapping Transcripts	78.29%
Align	Hisat	MUTANT	MT3	Mapping AS Genes	51.03%
Align	Salmon	MUTANT	MT3	Quantify AS Genes transcripts	60.7%

Gene repertoire: Table similar to a GFF for quick access to structural information of genes and mRNAs.

#	Name	Isoforms	Chromosome	Strand	Start	End	Size
1	AT1G02145	4	NC_003070.9	5'	404600	408619	4020
2	AT1G06620	2	NC_003070.9	5'	2025544	2027413	1870
3	AT1G07700	4	NC_003070.9	5'	2379595	2381390	1796
4	AT1G09140	3	NC_003070.9	3'	2942538	2945976	3439
5	AT1G09195	15	NC_003070.9	3'	2967940	2971568	3629
6	AT1G14700	4	NC_003070.9	5'	5058583	5061140	2558
7	AT1G26440	8	NC_003070.9	3'	9143937	9146106	2170
8	AT1G30540	4	NC_003070.9	3'	10816616	10819419	2804
9	AT1G31175	2	NC_003070.9	5'	11140471	11142716	2246
10	AT1G31950	3	NC_003070.9	5'	11475623	11478548	2926
11	AT1G33270	2	NC_003070.9	3'	12068063	12070455	2393
12	AT1G48210	7	NC_003070.9	5'	17798431	17802330	3900
13	AT1G62800	3	NC_003070.9	3'	23253703	23257546	3844
14	AT1G63900	2	NC_003070.9	5'	23716643	23719337	2695
15	AT1G73600	2	NC_003070.9	5'	27669152	27673572	4421
16	AT1G75420	3	NC_003070.9	5'	28305218	28307548	2331
17	AT1G77290	3	NC_003070.9	5'	29038366	29040217	1852
18	AT1G78200	5	NC_003070.9	5'	29419466	29421922	2457

Graphics: Graphics of the overall quantitative analysis and gene structure.



Events: Page containing information about the AS events identified in the analyses.
AS details: Table containing various information for identified AS events.

StartNewOverviewEventsGenesAbout

Detalhes de AS

Tabela de anotação

Graficos

Search gene

7702

Gene

Isoform1

Isoform2

Δ PSI

FDR

Δ Exonic Size

Δ CDS Size

Log2FC

Type

Maser

Event

min mRNA TPM

max mRNA TPM

min Protein Size

PTC

Chromosome

MUTANT gene TPM

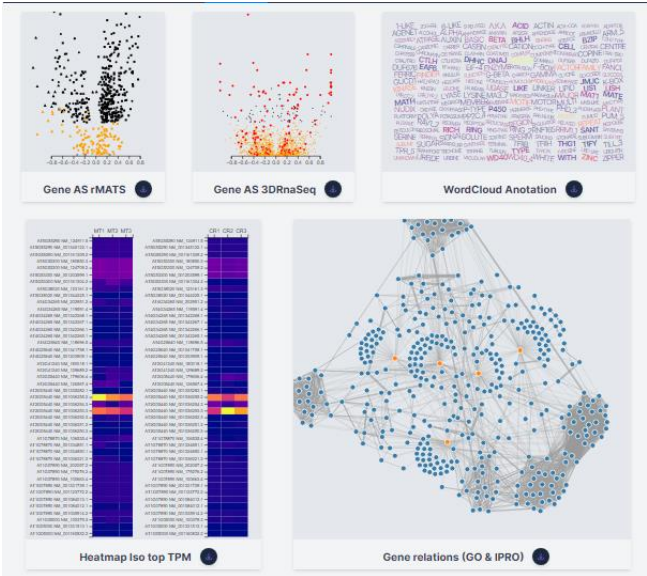
WILD gene TPM

#	Gene	Δ PSI	FDR	Δ CDS Size	Log2FC	Type	Event	min Protein Size
1	AT1G02145	0.41	0.00426	132	0	RI	2703	454
2	AT1G02145	0.41	0.00426	120	0	RI	2703	454
3	AT1G02145	0.41	0.00426	495	0	RI	2703	333
4	AT1G02145	0.41	0.00426	483	0	RI	2703	333
5	AT1G06620	-0.125	0.0223	243	0	RI	3823	285
6	AT1G09140	-0.179	0.0197	60	0	RI	2707	249
7	AT1G09140	-0.179	0.0197	24	0	RI	2707	249
8	AT1G09140	0.008	1.00	60	0	RI	2708	249
9	AT1G09140	0.008	1.00	24	0	RI	2708	249

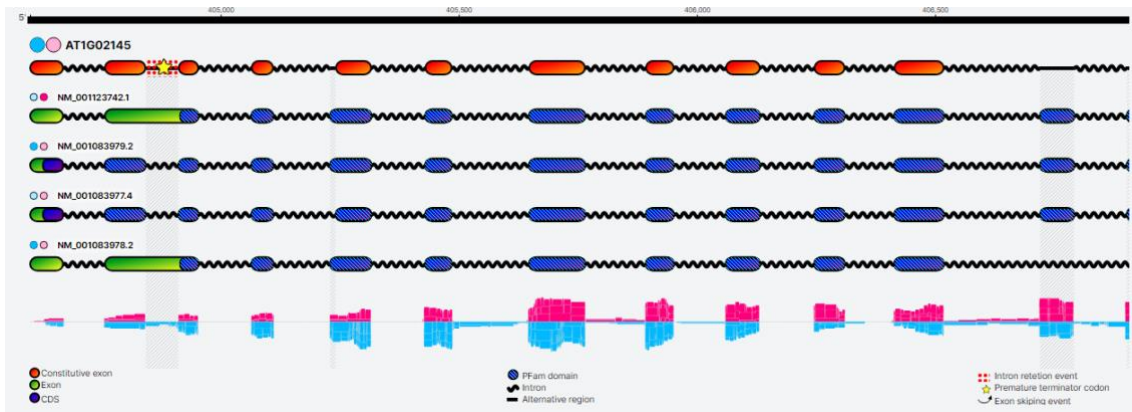
Annotation table: Table of gene annotation containing ontology, InterPro description, and identified Pathways.

Gene	Δ PSI	FDR	Gene Ontology	InterPro	Pathways
AT1G02145	0.410	0.00426	glycosyltransferase activity (GO:0016757)	GPI mannosyltransferase	PWY-1901, PWY-1961, PWY-1981... +202
AT1G09140	-0.242	0.0375	RNA binding (GO:0003723)	RNA recognition motif domain	PWY-102, PWY-1061, PWY-1187... +581
AT1G09140	-0.179	0.0197	RNA binding (GO:0003723)	RNA recognition motif domain	PWY-102, PWY-1061, PWY-1187... +581
AT1G09140	0.00800	1.00	RNA binding (GO:0003723)	RNA recognition motif domain	PWY-102, PWY-1061, PWY-1187... +581
AT1G14700	0.545	0.0399	hydrolase activity (GO:0016787)	Calcineurin-like phosphoesterase domain, ApaH type	PWY-4702, PWY-5381, PWY-5461... +445
AT1G26440	0.0560	1.00	membrane (GO:0016020)	Ureide permease	
AT1G26440	0.0560	1.00	membrane (GO:0016020)	Ureide permease, plant	
AT1G26440	0.0560	1.00	nitrogen compound transport (GO:0071705)	Ureide permease	
AT1G26440	0.0560	1.00	nitrogen compound transport (GO:0071705)	Ureide permease, plant	
AT1G26440	0.0560	1.00	transmembrane transporter activity (GO:0022857)	Ureide permease	
AT1G26440	0.0560	1.00	transmembrane transporter activity (GO:0022857)	Ureide permease, plant	
AT1G26440	-0.0450	1.00	membrane (GO:0016020)	Ureide permease	
AT1G26440	-0.0450	1.00	membrane (GO:0016020)	Ureide permease, plant	
AT1G26440	-0.0450	1.00	nitrogen compound transport (GO:0071705)	Ureide permease	

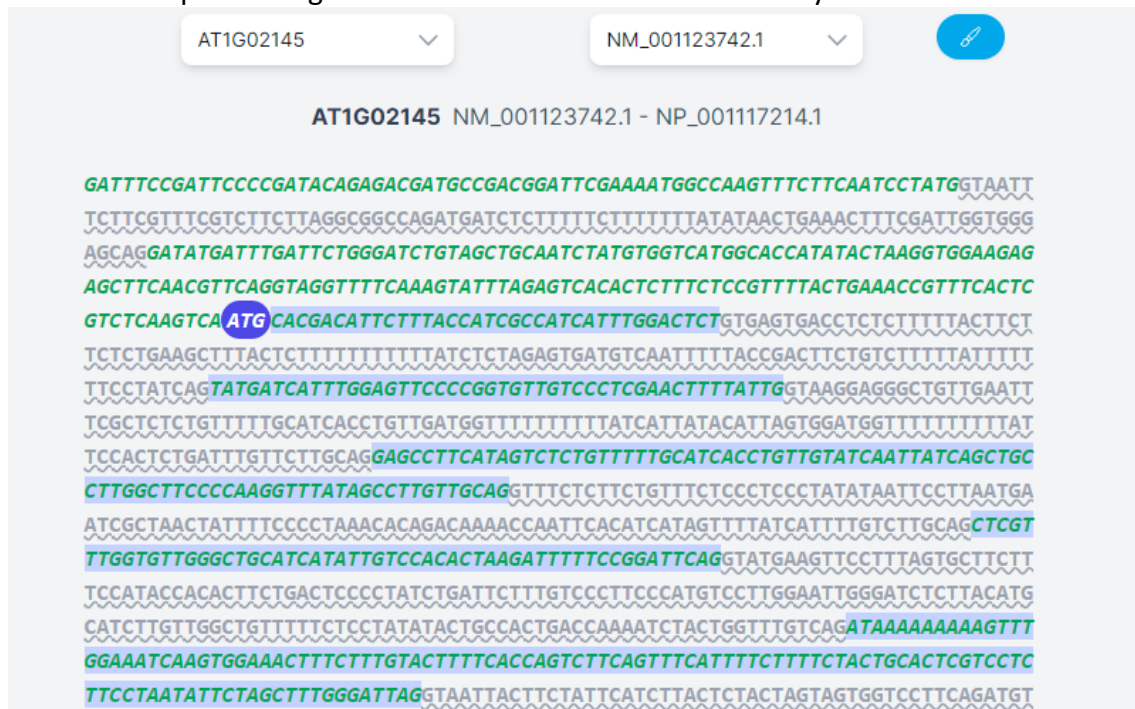
Graphics: Graphics of AS events, annotation, and differential expression.



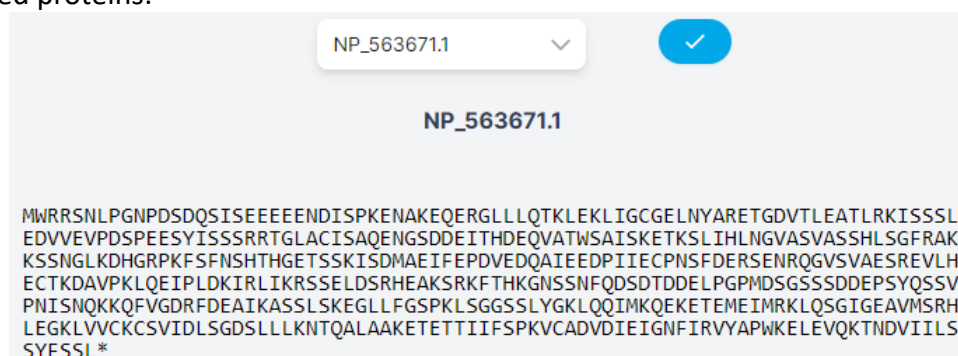
Genes: Page that displays gene by gene its structure and functional annotation.



Genomic: Sequence of gene nucleotides that can be colored by exons and CDS



Protein: Sequence of amino acids that can be annotated by UniProt to explore similar mapped proteins.



How to extract the information?

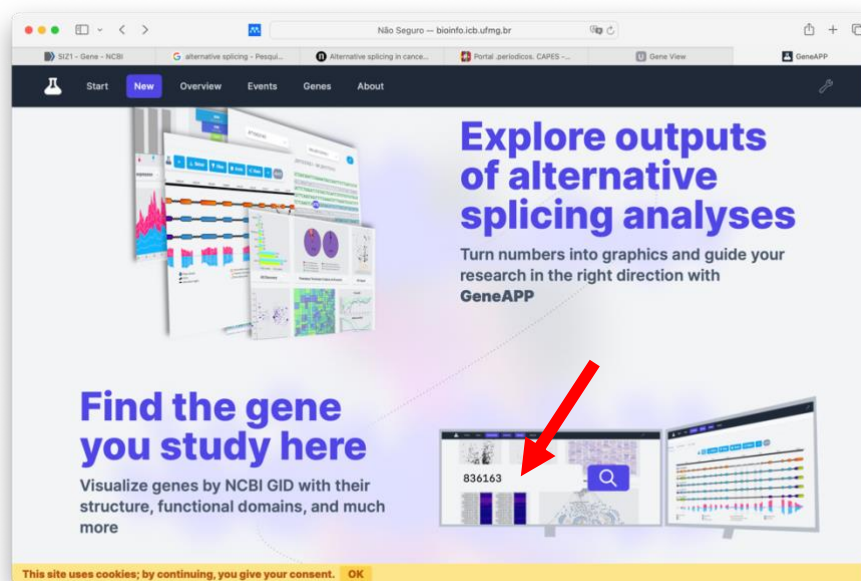
The user has the option to download the results or share the gene along with annotations of identified AS events. The files available for exploration in the GeneAPP test analysis can be obtained by clicking the button. Figures are generated in vector format, enabling high-quality print rasterization. Tables are exported in TSV format.

3. Use cases of Geneapp

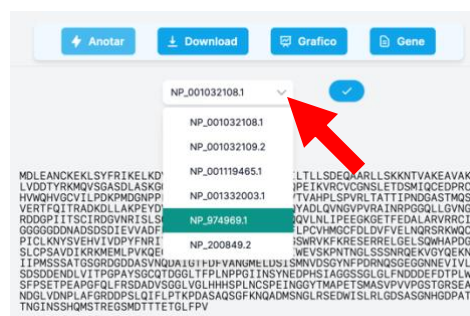
3.1. UC1: Consulting the gene structure in GeneappExplorer

In this case, the user wants to visualize the gene structure of SIZ1 graphically using GeneappExplorer. Ultimately, the user wants to download the graphical representation of the gene as a vector figure.

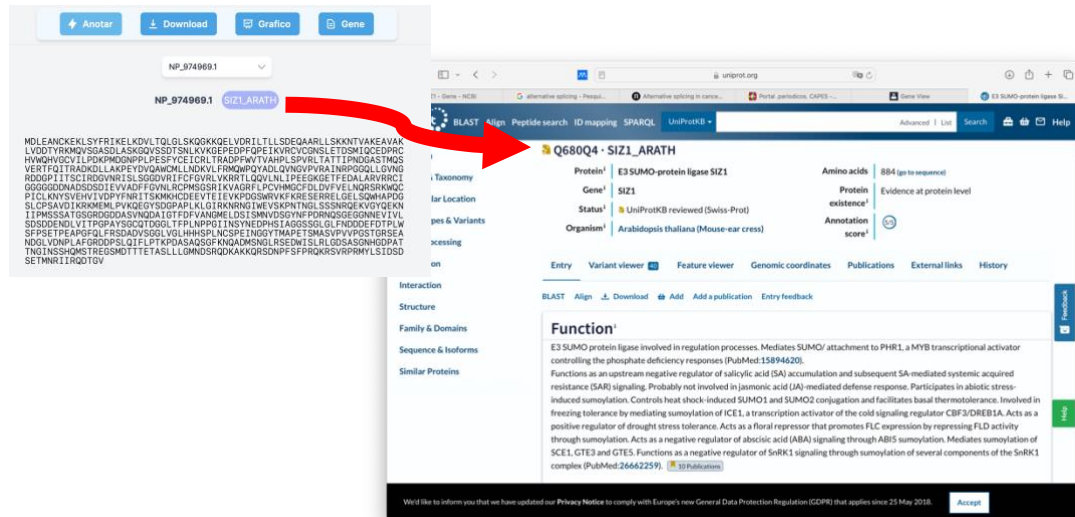
1. The user locates the GeneID by using the search box in the NCBI Gene database: <https://www.ncbi.nlm.nih.gov/gene/?term=SIZ1>, 836163
2. The user accesses <http://bioinfo.icb.ufmg.br/geneapp> and, at the location indicated by the red arrow in the figure below, enters the GeneID and presses the “search” button to start.



3. On the opening page, the user presses the button **Anotar** to mark protein domain families on the gene. The user will be prompted to enter their email to access the InterproScan API. After a few minutes, the figure will be redrawn.
4. The user presses **Download** to download the figure with annotation. After that, click on the button **Sequencia** to visualize details for the gene sequence, and click on **Proteina** to access the protein sequence.
5. The user selects the protein of interest from the dropdown menu indicated by the arrow in the figure below and clicks on **✓** to identify similar proteins on Uniprot.



- Next, the ID of the similar protein in UniProt will be displayed, as shown in the figure on the left below. The user clicks on the ID and is redirected to UniProtKB, where they have access to various annotations of similar proteins, as depicted in the figure on the right below.



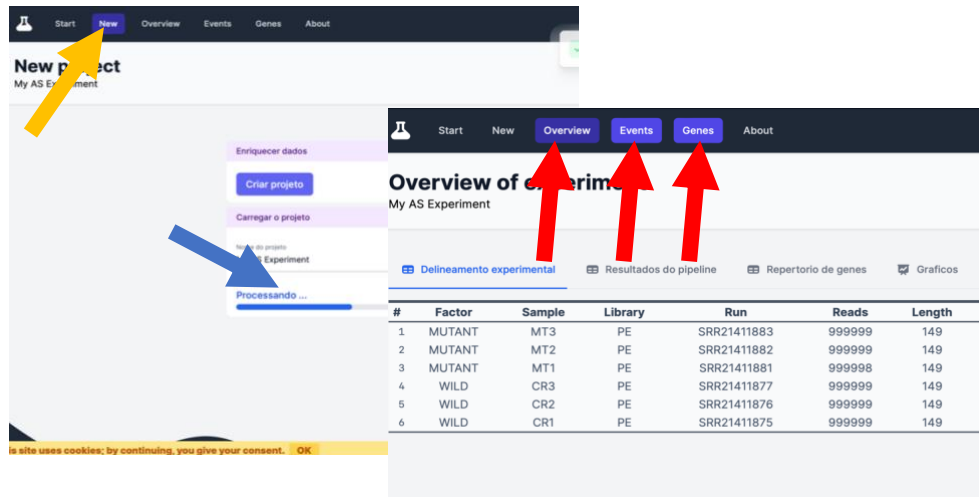
3.2. UC2: Performing a DAS analysis with GeneappScript

In this use case, the user has their experiment defined: genome, transcriptome, proteome, and RNA-seq triplicates obtained from wild-type and mutant samples of Arabidopsis roots. The user wants to identify DAS genes by comparing the experimental groups using GeneappScript.

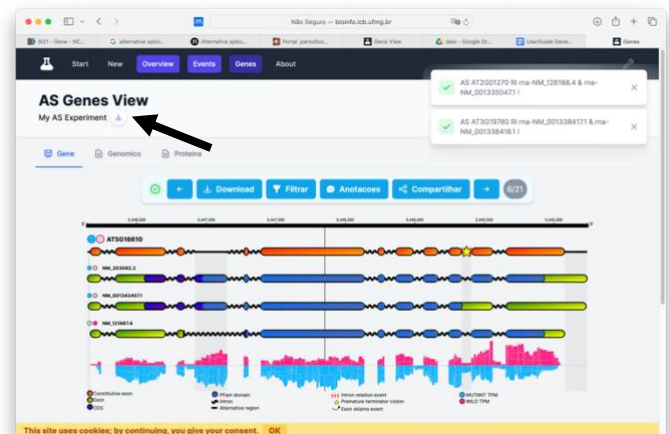
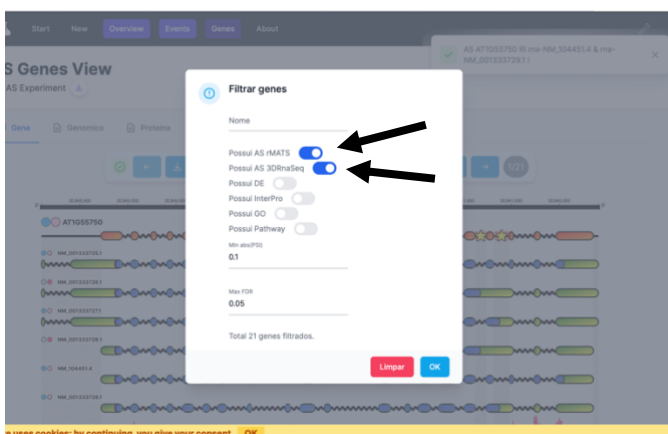
- The user accesses the Colab on <https://colab.research.google.com> and inserts the command below to execute the GeneAPPScript build based on your experimental design:

```
! curl -s https://raw.githubusercontent.com/MiqueiasFernandes/GeneAPP/main/public/data/as_data_gen.sh | \
  bash -s - \
  'https://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/001/735/GCF_000001735.4_TAIR10.1/GCF_000001735.4_TAIR10.1_genomic.fna.gz' \
  'https://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/001/735/GCF_000001735.4_TAIR10.1/GCF_000001735.4_TAIR10.1_genomic.gtf.gz' \
  'https://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/001/735/GCF_000001735.4_TAIR10.1/GCF_000001735.4_TAIR10.1_cds_from_genomic.fna.gz' \
  'results' \
  SRA_ARG1='--minReadLen' SRA_ARG2='150' SRA_ARG3='-X' SRA_ARG4='999999' \
  SRR21411875,CR1,WILD SRR21411876,CR2,WILD SRR21411877,CR3,WILD \
  SRR21411881,MT1,MUTANT SRR21411882,MT2,MUTANT SRR21411883,MT3,MUTANT \
  RMATS_ARG1='--readLength' RMATS_ARG2='150' GEN_NCBI_TABLE KEEP_TMP \
  1> log.txt 2> err.txt
```

- At the end of the process, which takes approximately 6 hours (about 1 hour per sample), the user downloads the files with the name 'part***.geneapp' that were generated in the folder results/import_geneapp.
- The user then accesses GeneappExplorer at <http://bioinfo.icb.ufmg.br/geneapp> and on the 'New' page, indicated by the yellow arrow in the figure below, uploads the files downloaded via the button located in the region indicated by the blue arrow. After a few seconds, when the file processing is complete, the pages indicated by the red arrows in the following figure will become available to the user.



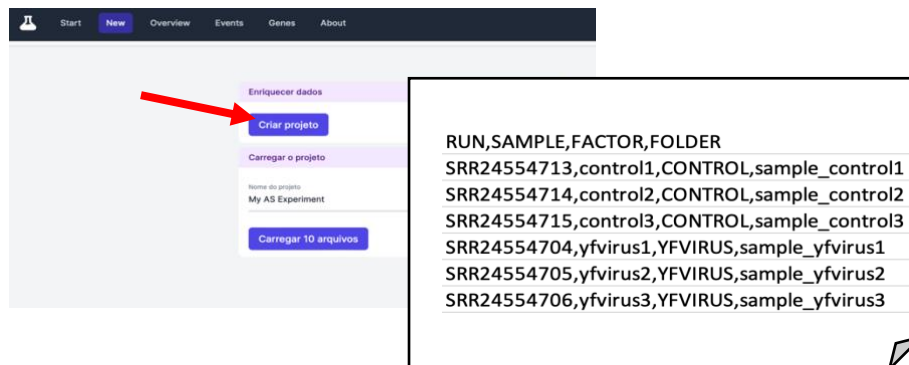
- On the first page, the user has access to descriptive tables and graphs of the experiment. On the 'Events' page, the user can access tables and graphs related to the identified DAS events and graphical representations of the functional annotation of the proteins.
- On the 'Genes' page, the user has access to the graphical structure of the genes, displaying the AS event under the gene structure and the read coverage support. The user wishes to filter for genes with evidence from both rMATs and 3DRNaseq software. Using the button **Filtrar**, the user checks the switches indicated by the arrows in the figure on the left below, exploring only the 21 filtered genes. To download all generated and visualized results, the user clicks on the download button indicated by the arrow in the figure on the right below.



3.3. UC3: Exploring data from another software

The user already has all the output tables generated by rMATs and 3DRNaseq, or they have tables in a specific format generated by other programs like AStalavista, and they want to explore these results in GeneappExplorer. If the tables are not in the rMATs or 3DRNaseq format, the user converts them to one of these two standards, depending on whether the software is event-oriented or isoform-oriented.

1. The user accesses the GeneappServer interface on <http://bioinfo.icb.ufmg.br/geneapp/new> and clicks on the button indicated in the figure below on the left to upload their *experimental_design.csv* file as shown in the figure below on the right.



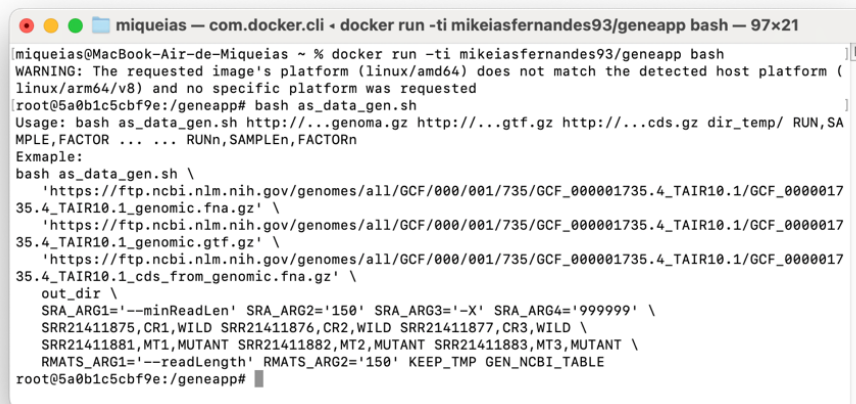
2. The user is prompted to upload the remaining files they have via a dialog window. After processing, which takes a few minutes, the files will be automatically downloaded.
3. The user then explores the results after refreshing the page and proceeding according to UC2 from step 3 onwards.

3.4. UC4: Using Geneapp via Docker

The user can download the source code of Geneapp and run it on their computer. Installing the necessary environment to run GeneappScript may take about 2 hours due to the large number of R packages from the dependencies MASER and 3DRNASEQ. Docker brings the convenience of GeneappScript in a portable and pre-configured environment, ready for execution. To set up GeneAPP via Docker, the user:

1. Installs the Docker on the host machine as follows on <https://docs.docker.com/desktop/>
2. Runs the command `docker pull mikeiasfernandes93/Geneapp`
3. Runs the analysis of GeneappScript with the command ***docker run -ti mikeiasfernandes93/geneapp bash***

4. Inside the folder /geneapp on the virtual machine, the user can run the command *bash as_data_gen.sh* and follow the instructions in the figure below



```
mikeias@MacBook-Air-de-Miqueias ~ % docker run -ti mikeiasfernandes93/geneapp bash
WARNING: The requested image's platform (linux/amd64) does not match the detected host platform (linux/arm64/v8) and no specific platform was requested
[root@5a0b1c5cbf9e:/geneapp# bash as_data_gen.sh
Usage: bash as_data_gen.sh http://...genoma.gz http://...gtf.gz http://...cds.gz dir_temp/ RUN,SAMPLE,FACTOR ... .. RUNn,SAMPLEn,FACTORn
Exmaple:
bash as_data_gen.sh \
'https://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/001/735/GCF_000001735.4_TAIR10.1/GCF_000001735.4_TAIR10.1_genomic.fna.gz' \
'https://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/001/735/GCF_000001735.4_TAIR10.1/GCF_000001735.4_TAIR10.1_genomic.gtf.gz' \
'https://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/001/735/GCF_000001735.4_TAIR10.1/GCF_000001735.4_TAIR10.1_cds_from_genomic.fna.gz' \
'out_dir' \
SRA_ARG1=\\'--minReadLen\\' SRA_ARG2=\\'150\\' SRA_ARG3=\\'-X\\' SRA_ARG4=\\'999999\\' \
SRR21411875,CR1,WILD SRR21411876,CR2,WILD SRR21411877,CR3,WILD \
SRR21411881,MT1,MUTANT SRR21411882,MT2,MUTANT SRR21411883,MT3,MUTANT \
RMATS_ARG1=\\'--readLength\\' RMATS_ARG2=\\'150\\' KEEP_TMP GEN_NCBI_TABLE
root@5a0b1c5cbf9e:/geneapp#
```

4. Outputs of Geneapp

After running the GeneappScript, various temporary files, logs, output files from intermediate programs, and files generated by Geneapp are stored in the output directory. These files are organized into two folders: 'out_dir' and 'temp_***', with tables in CSV format and figures in PNG, SVG, or PDF format. In GeneappExplorer, all tables and graphs presented can be downloaded in CSV or SVG format with a transparent background, as summarized in the following topics:

- **Tables**
 - **Experiment_table:** Experimental design
 - **Pipeline_table.csv:** Results overview
 - **Gene_table.csv:** Genes structure in GFF layout
 - **Events_table.csv:** Quantitative analysis of DAS
 - **Annotation_table.csv:** Gene annotation
- **Descriptive graphs**
 - **graphQc.svg:** Sample quality control graph
 - **graphRd.svg:** Genome mapping, transcripts, and AS genes graph
 - **graphGc.svg:** Graph of the genomic structure of regions where DAS occurs
 - **graphUp.svg:** Graph of expressed genes per sample
 - **graphCv.svg:** Graph of mapping depth across genes per sample
 - **graphAs.svg:** Graph of gene filtering steps
 - **graphVen.svg:** Graph of shared genes in DE and DAS analysis
 - **graphMp.svg:** Graph of DAS analysis per software
- **Analysis graphs**
 - **graphBar.svg:** Graph of identified event types
 - **graphTop.svg:** Graph of top 10 PSI gene
 - **graphPie.svg:** PTC graph
 - **graphCov.svg:** Graph of mapping depth at critical AS sites

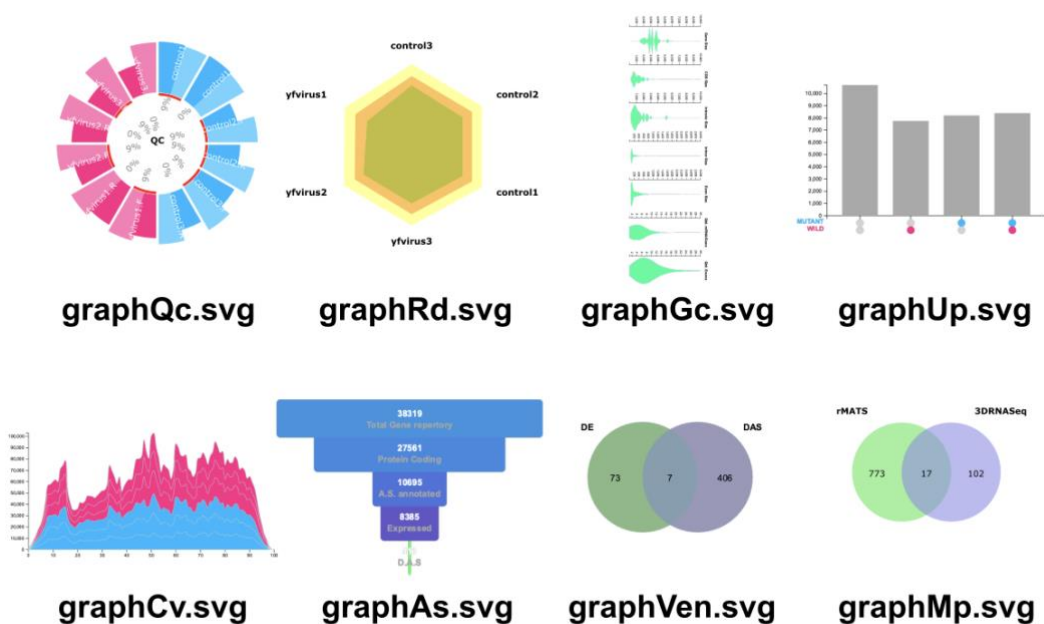
- **graphDea.svg**: DEG with DAS integration graph
- **graphFi.svg**: Graph of clustering of DAS genes by sequence
- **Volcano plots**
 - **graphDe.svg**: Graph of differential expression
 - **graphDa.svg**: DAS of 3DRNASeq graph
 - **graphDar.svg**: DAS of rMATS graph
 - **graphScr.svg**: Graph of the impact of AS changes on the gene
- **Heatmaps**
 - **graphHm.svg**: Graph of gene expression
 - **graphHm2.svg**: Graph of transcripts expression
- **Integration graphs**
 - **graphAn.svg**: Annotation graph
 - **graphTr.svg**: Graph of functional domain sizes in the gene
 - **graphWc.svg**: Graph of annotation keywords
 - **graphLk.svg**: Graph of gene relationships

4.1. Figures generated by other software

The graphs and tables generated by rMTAS and MASER are stored in the 'out_dir/rmats_out' folder, with tables for event-specific results filtered by read coverage in the 'coverage_filt' folder. The output files from 3DRNASeq are stored in the 'out_dir/to3d' folder, including the grouped graphs in the 'Rplots.pdf' file generated within this folder. At the end of the GeneappScript execution, the MultiQC report 'multiqc_report.html' is generated in 'out_dir/multiqc'. This report is an HTML page containing a series of graphs that can be extracted on demand.

4.2. GeneappExplorer Descriptive graphs

These graphs generated in GeneappExplorer serve to describe the data used in the analyses, namely: graphQc.svg, graphRd.svg, graphGc.svg, graphUp.svg, graphCv.svg, graphAs.svg, graphVen.svg, and graphMp.svg. The following image shows examples of them.



In the circular graph `graphQc`, each sample is represented by three bars, integrating the main metrics of the raw data. The innermost bar represents the rate of reads that did not pass the quality control of Trimmomatic, the darker bar represents the number of reads in the raw sample, and the lighter bar represents the average read size in the sample, according to the colors of the experimental group.

In the radar graph `graphRd.svg`, each sample is represented according to the mapping rate reported by the Hisat2 software using the 'overall alignment rate' metric. Moving from inside to outside: mapping rate in transcripts in brown, in genes in orange, and the genome in yellow. This graph is useful for showing that the RNA-seq samples are compatible with the reference genome and annotated genes in the genome.

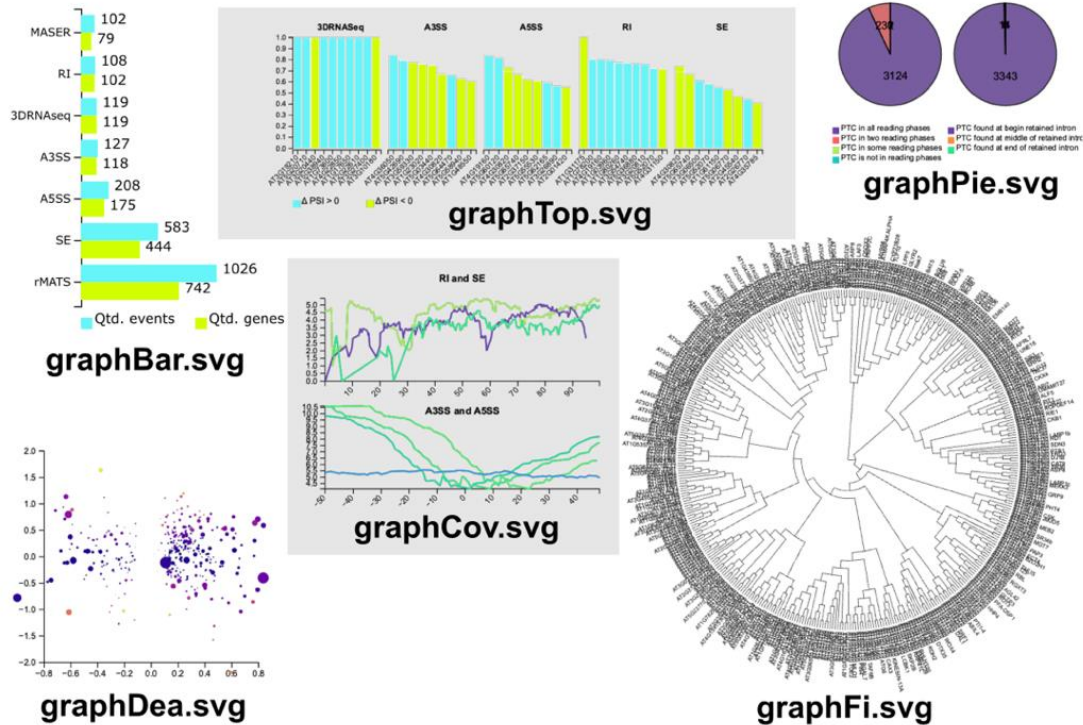
The violin plot `graphGc.svg` depicts the structure of DAS genes and their elements in terms of the size of genes, CDS, total introns in the gene, individual introns, and exons. Violin plots are also presented regarding the count of transcripts and exons per gene.

The Upsetplot `graphUp.svg` shows the number of genes qualitatively related to their expression (if $TPM > 0$) in the samples, grouped by experimental group. In the area plot `graphCv.svg`, genes are presented based on their expression in each sample, from 0% to 100% of their sequence on the X-axis, and the read count on the Y-axis.

The funnel plot `graphAs.svg` displays the number of genes that have progressed to the next stage of the pipeline from the total number of genes in the GFF file to the quantity identified with DAS. The Venn diagram `graphMp.svg` separates the DAS genes by software, while `graphVen.svg` separates them by DEG or DAS approach. These graphs provide an overview of the pipeline and how the data were processed throughout it.

4.3. Analysis chart

The analysis graphs provide an overview of DAS analysis through the following plots: graphBar.svg, graphTop.svg, graphPie.svg, graphCov.svg, graphDea.svg, graphFi.svg as shown in the figure below.



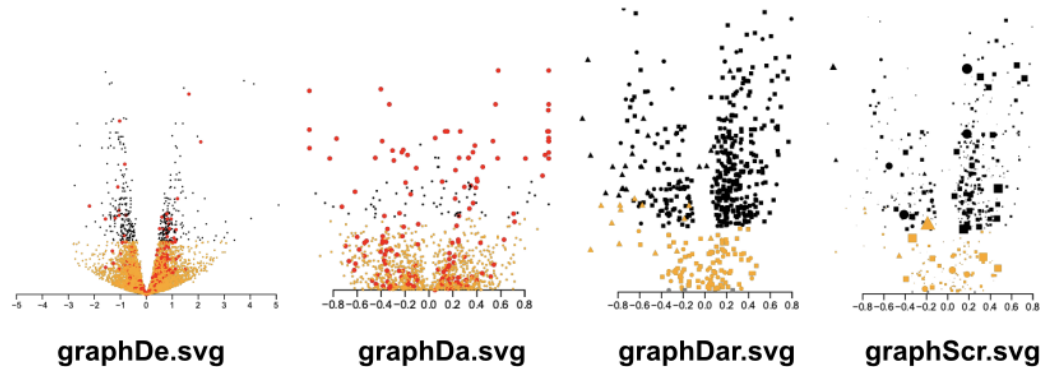
In the bar graph graphBar.svg, the quantities of events and genes identified by program and event type are presented. In the graph graphTop.svg, the genes with the highest absolute difference in PSI between the compared experimental groups are shown. The pie chart graphPie.svg presents the PTC rate identified in RI events, indicating both the quantity (whether in all, 2, 1, or none of the reading phases) and the position (whether at the beginning, middle, or end) where it occurs in the intron. Since a retained intron event can introduce a PTC, PTCs are expected to be present in all reading phases and between the beginning and middle of the retained intron in these graphs. Additionally, the line graph graphCov.svg displays read coverage from 0% to 100% of the alternative region in RI and SE, and 50 bp upstream and downstream of the alternative site in A3SS and A5SS. This graph presents genes with the highest TPM, with a constant line expected in RI and SE and a valley formation for A3SS and A5SS, as demonstrated in the example figure.

The scatter plot graphDea.svg integrates DEG results with DAS, placing ΔPSI on the X-axis and $\log_2(FC)$ on the Y-axis. The size of the point indicates the impact of the AS event on the gene, and the color indicates the statistical significance (FDR) of the AS event.

To assist researchers in forming groups of genes in the repertoire of DAS genes identified by the programs, the graphFi.svg presents a clustering based on gene sequences. The graph is constructed using the web service <https://www.genome.jp/tools-bin/ete> using the mat-default parameters.

4.4. GeneappExplorer Volcanoplot

Four volcano plots are generated: graphDe.svg, graphScr.svg, graphDa.svg, and graphDar.svg as demonstrated in the figure below.



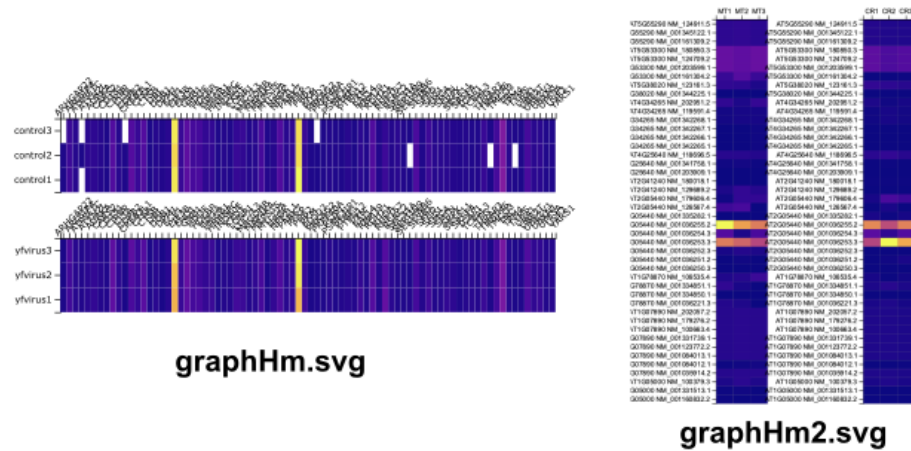
The graph graphDe.svg presents DEG genes, where each point represents a gene. Points in black are statistically significant, and those in red indicate genes with DAS identified by some program. In all four graphs, the Y-axis scales statistical significance (p-value), while the X-axis represents Log2(FC) in graphDe.svg and Δ PSI in the others.

The graph graphDa.svg displays DAS genes identified by 3DRNASEQ, with genes identified by rMATS colored in red. In graph graphDar.svg, DAS events found by rMATS are represented: SE events are shown as squares, RI as triangles, and A3SS and A5SS as circles.

Graph graphScr.svg follows the same logic as the previous one, with the size of the shapes proportional to the impact of the AS event on the normalized gene. For example, given two genes under DAS: g1 has a size of 100 and an SE event in an exon of size 50, and g2 has a size of 200 and an RI event in an intron of size 10. The impact will be 50% for g1 and 5% for g2, resulting in a shape 10 times larger for g1 compared to g2 in the graph.

4.5. GeneappExplorer Expression heatmap

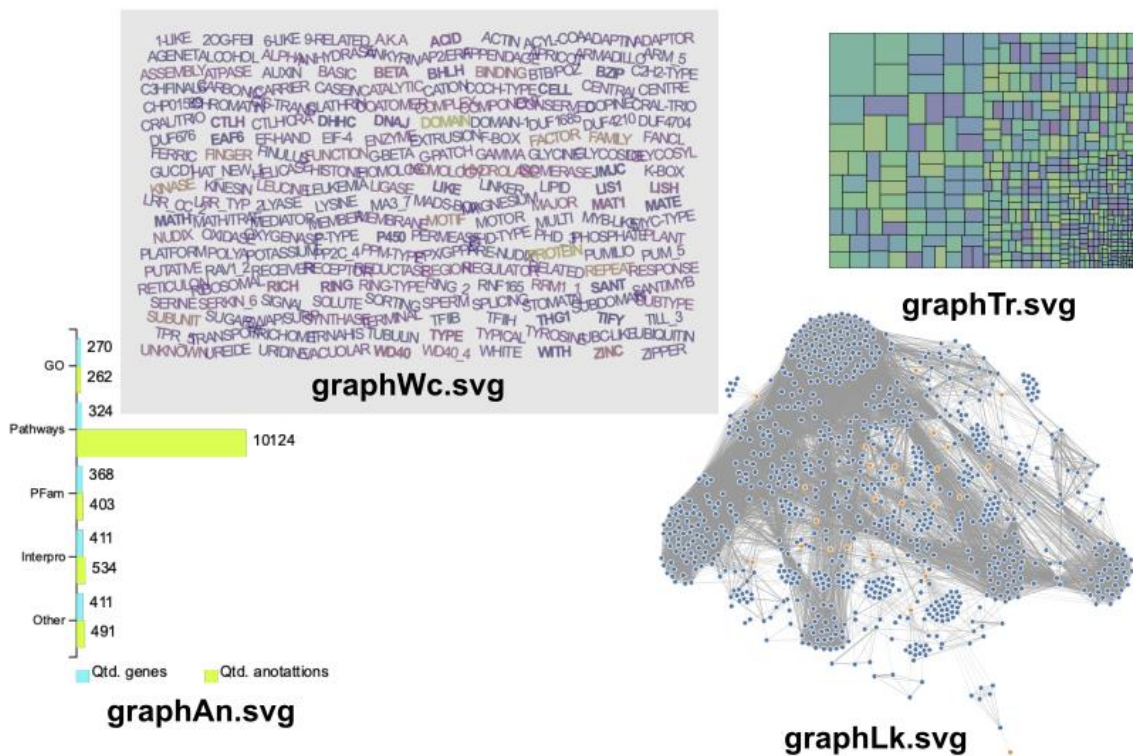
With the TPM calculated by 3DRNASeq, two heatmaps are generated: graphHm.svg and graphHm2.svg as shown in the figure below. In the graphs, the elements are grouped by sample and experimental group.



In the horizontal heatmap graphHm.svg, genes with the highest TPM and greatest DAS are presented, while in the graph graphHm2.svg, isoforms are presented following the same filter.

4.6. GeneappExplorer Functional annotation chart

With the functional annotation data of the proteins generated by InterproScan5, four graphs are presented: graphAn.svg, graphLk.svg, graphTr.svg, graphWc.svg as illustrated in the figure below. These graphs aim to contextualize the transcriptomic level explored in the previous graphs with the proteomic level of the identified DAS genes.



In the graph graphAn.svg, annotations grouped by major biological databases integrated by GeneAPPScript are quantitatively summarized. This graph indicates that the annotation will be useful when its quantity exceeds the quantity of genes.

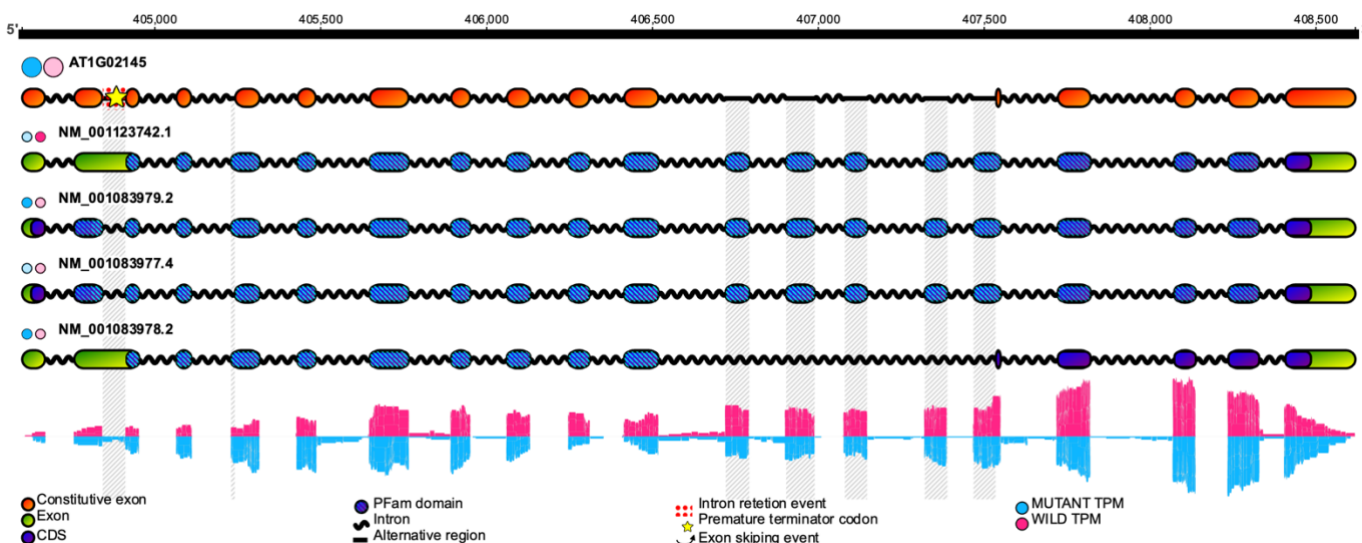
In the word cloud of the graph graphWc.svg, the most frequent words in the annotations – protein names, gene ontology terms, names of protein domain families (PFAM), Interpro annotation names, etc. – are presented with larger size and more prominent color.

In the graph graphTr.svg, regions identified by Interpro are counted in each rectangle, and the largest rectangle represents the site with the largest size in the proteome of DAS genes, summing all its occurrences in the dataset. Hovering over the Interpro ID rectangle displays its details, and clicking on the rectangle redirects the user to the Interpro website to view all the details of the record.

The graph graphLk.svg is useful for identifying groups of genes in the repertoire of DAS genes. In this graph, each blue dot represents a gene, and the orange dots represent the chromosomes. Genes are connected by common functional annotation, such as GO or PFam. If a gene is located on the genome within 10Kbp of another, the vertex between them is presented thicker in the graph. Genes with many common connections come closer together, forming clusters in the graph. The graph is interactive, allowing users to rearrange the dots by holding and dragging them for better visualization.

4.7. GeneappExplorer AS event structure graph

On the genes page, the user has access to the individual visualization of each gene, allowing the presentation of the gene's structure, its isoforms, the AS event, and contextualizing the genomic, transcriptomic, and proteomic levels, as shown in the example figure below.



In this graph, the user is presented with a scale at the top based on the genome coordinate where the gene occurs, and the strand orientation 5'-3' (sense) or 3'-5' (anti-sense). If the user moves the mouse over the scale, a vertical line will be displayed to

facilitate inspection of the gene's structure. The position value will also be displayed on the line, starting from 1bp on the left.

The circles preceding the gene and isoform names are filled with intensity according to the mean TPM presented in each experimental group as per the legend. Through them, the user can identify which isoforms were altered in the occurrences identified by 3DRNASeq and evaluate the types of events that occurred. As the graph contextualizes both 3DRNASeq and rMATS programs, in this example graph, one can identify that the event identified by 3DRNASeq, for example, is of the RI type as seen in the second intron of the gene structure, with the star indicating the occurrence of PTC in the intron.

Below the gene name, a consensus structure of its isoforms is presented, with alternative regions in dash format, constitutive introns in wavy lines, and exons in rectangular orange shapes. This structure is used to present the structure of the AS event identified by rMATS and facilitates the user in identifying the parts of the gene involved in the event. In the figure, for example, the RI event is caused by the retention of the second intron (based on where the event occurred) between the first two isoforms presented (based on the filled circles before the gene name).

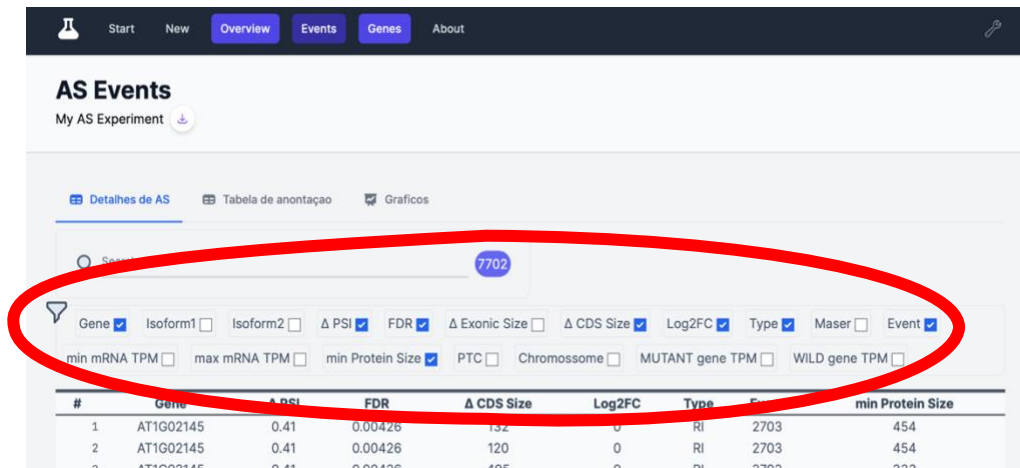
The expression in TPM of each replicate of the experimental groups is presented stacked in the graph at the bottom of the figure. In the isoforms, the coding DNA sequences (CDS) are colored in purple, and occurrences of PFam functional domains identified by InterproScan are presented as light blue stripes in the CDS, allowing the user to contextualize the genomic, transcriptomic, and proteomic levels.

4.8. Tables

The GeneappExplorer provides 5 tables for download:

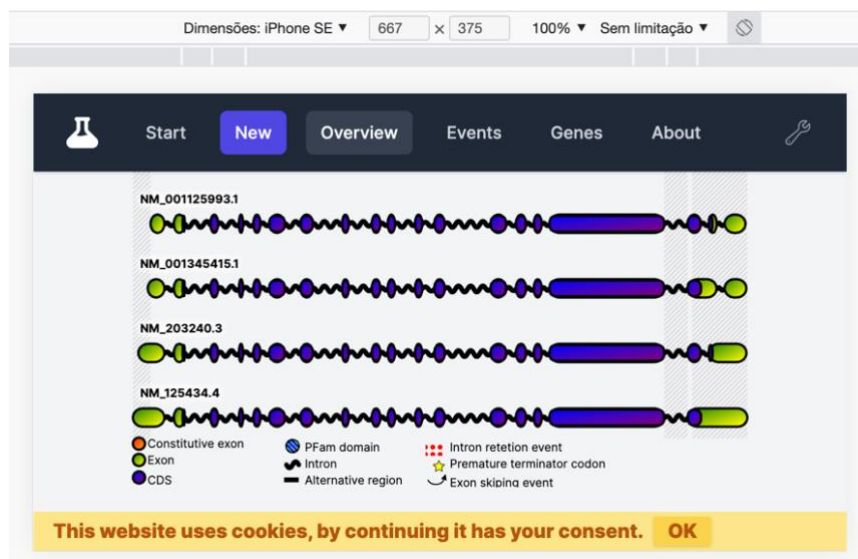
- **Experiment_table.csv:** This table provides an overview of the dataset used by GeneappScript, relating the experimental groups and their labels, along with some metadata about the dataset.
- **Pipeline_table.csv:** This table presents numbers and statistics calculated by GeneappScript for each stage recorded in its log file produced during execution. It is very useful for users to quantitatively track the outputs of the software at each stage of the pipeline.
- **Gene_table.csv:** This table informs about the structure of DAS genes in a layout similar to GFF3.
- **Annotation_table.csv:** This table provides annotations of DAS genes in columns such as Gene, Δ PSI, FDR, Gene Ontology, InterPro, and Pathways.

- **Events_table.csv:** This table presents each event identified by contextualizing data from 3DNRASeq with data from rMATS. The columns vary among 18 available options, which are enabled according to the options selected by the user in GeneappExplorer. Some default columns include Gene, Δ PSI, FDR, and Log2FC. The available columns are presented in the highlighted region in the figure below.



5. Developers information

The Geneapp was also tested in a mobile environment, and the GeneappExplorer application was able to adjust and plot the figure on devices with small screen sizes. The simulation can be viewed in the following figure.



In case of any bugs, please report them at <https://github.com/MiqueiasFernandes/GeneAPP/issues>, indicating the environment where the problem occurred or the issue to be resolved. For contacting the authors and developers regarding questions, suggestions, or compliments, please use geneapp23@gmail.com.

6. About Geneapp

Geneapp (2022, Miquéias Fernandes and Edson Mario de Andrade Silva) emerged from various scripts that were necessary to write to explore the outputs of software for the identification of differential AS. Some of these scripts are in the repository Bioinformatics and the generated notebooks for a coffee analysis. During the analysis of DAS with plant, mosquito, and fungus datasets, our research group accumulated scripts to process, integrate, and visualize tables generated by DAS identification software. Processing capabilities included, for example, filtering raw results and extracting alternative isoform sequences. Integration refers to merging results from different software, such as rMATS outputs with 3DRNASeq outputs or a General Feature Format (GFF) file with InterproScan tabular output. When exploring the results, there was often a need to generate graphs to visualize the AS event under the gene or a group of their occurrences in a scatter plot, bar graph, or heatmap. This stack of scripts became the core of Geneapp, which was designed to meet these needs. For the development of the tool, the problem was divided into three independent parts so that the tool could be better used. GeneappScript was developed to generate integrated and processed tabular results, GeneappServer to parse results that the researcher already has into the Geneapp format, and GeneappExplorer to plot the graphs and explore the results. The development of the application was carried out in 2022 on the GeneappScript, GeneappServer, and GeneappExplorer modules following the success of the web app prototype made available in 2021 for testing.

Revised Geneapp: Édson M. A., Gustavo, and Bruno. Geneapp uses resources from various developers, credits to:

- [NCBI](#) provided biological data and programs
- [EMBL](#) provided biological data and programs
- [rMATS](#) generates input data
- [3DRNASeq](#) generates input data
- [Deeptools](#) calculates read depth
- [MASER](#) filters significant AS events
- Gimp-generated images for the home page
- [Inkscape](#) graphic design
- [Colab](#) infrastructure to run the test analysis
- [Firebase](#) WebAPP hosting
- [D3](#) plotting of graphics
- [Vue](#) APP's JS framework
- [Tailwind](#) APP's CSS framework
- [heroicons](#) app icons
- [d3-graph-gallery](#) app graphics
- [Observablehq](#) app graphics
- [Docker](#) app script container

7. License

Geneapp source code is open on <https://github.com/MiqueiasFernandes/GeneAPP>

Geneapp software is registered on INPI with number BR512023000902-4 to UNIVERSIDADE FEDERAL DE VIÇOSA; UNIVERSIDADE FEDERAL DE MINAS GERAIS. When using Geneapp, please be reminded that commercial use is not permitted and that the

software it executes (such as rMATS and 3DRNASeq) has specific licenses that must be observed. Additionally, keep in mind that the results provided by Geneapp need to be validated in the laboratory.

8. References

rMATS: <https://doi.org/10.1073/pnas.1419161111>
3DRNASeq: <https://doi.org/10.1080/15476286.2020.1858253>
Salmon: <https://www.nature.com/articles/nmeth.4197>
Hisat2: <https://www.nature.com/articles/s41587-019-0201-4>
MASER: <https://bioconductor.org/packages/maser>
MULTIQC: <https://doi.org/10.1093/bioinformatics/btw354>

9. Abbreviations

A3SS	Alternative 3' Alternative splicing site
A5SS	Alternative 5' Alternative splicing site
AI	Artificial Intelligence
AS	Alternative Splicing
BP	base pairs
CLI	Command line interface
DAS.....	Differential AS
DEG	Differentially expressed gene
ES	Exon skipping
GUI	Graphical user interface
MXE.....	Mutually exclusive event
NMD.....	Nonsense-mediated mRNA decay
PSI	Percent spliced in
PTC	Premature terminator codon